REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
19-02-2010	Final Report	12/01/2006 -11/30/2009

4. TITLE AND SUBTITLE
Geospatial representation, analysis and computing using bandlimited functions

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-07-1-0135

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Cory Ahrens, Gregory Beylkin and Kristian Sandberg

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Colorado
Department of Applied Mathematics
UCB 526
Boulder, CO 80309

8. PERFORMING ORGANIZATION REPORT NUMBER
1543425

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
875 North Randolph Street
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2012-0628

12. DISTRIBUTION/AVAILABILITY STATEMENT
A

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Space surveillance, navigation of aircraft and missiles require detailed representations of gravity and efficient methods for determining orbits and trajectories. However, many of the current mathematical representations and associated numerical algorithms were developed at the beginning of the computer era and are not efficient on today's computers. Under this grant new, computationally efficient, localized representations of gravity have been developed and tested. As a step in developing a new approach to estimating gravitational potentials, a multiresolution representation for gravity estimation has been proposed. New near optimal quadratures for integration on the sphere have been developed. These quadratures (with many additional applications) also yield a new method of interpolation on the sphere and will play a role in solving multiresolution estimation problem. New ODE integrators for orbit and trajectory calculations were developed and have been transferred to the Air Force. New integrators reduce the number of evaluations of the gravity forces and have improved stability while retaining accuracy and speed of the traditional integrators.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Gregory Beylkin
Unclassified				26	19b. TELEPHONE NUMBER *(Include area code)* 303 4926935

FINAL REPORT ON "GEOSPATIAL REPRESENTATION, ANALYSIS AND COMPUTING USING BANDLIMITED FUNCTIONS", AFOSR GRANT FA9550-07-1-0135

CORY AHRENS, GREGORY BEYLKIN AND KRISTIAN SANDBERG

Summary

The original goal of the proposal was to address three problem areas:

(1) Develop methods for representation and accurate interpolation of space-limited portions of data using bases for bandlimited functions.
(2) Develop numerical integrators for solving systems of the Ordinary Differential Equations (ODEs) that are based on quadratures for bandlimited functions, rather than polynomials.
(3) Develop representation of functions on the sphere, bandlimited functions on spherical patches and associated multiresolution bases.

We obtained new significant results in all three areas which were published in 3 papers and 1 preprint. Let us summarize these results.

(1) We have transferred code that generates a local approximation to gravity models to Brandon Jones, a Ph.D. student in Aerospace Department at CU, who made some improvement and fixed several bugs. The results of testing these models have been submitted as the paper B. Jones, G. Born and G. Beylkin, "Comparisons of The Cubed Sphere Gravity Model with the Spherical Harmonics" to Journal of Guidance, Control, and Dynamics, to appear in Volume 33, Number 2 (see **http://dx.doi.org/10.2514/1.45336**). During the summer of 2009 Brandon Jones implemented these localized gravity models for NASA at Johnson Space center.

We also developed a new approach to approximating solutions of Laplace's equation satisfying boundary conditions using Gaussians. Our new representation inherits a multiresolution structure from the Gaussian approximation, leading to fast algorithms for the evaluation of the solutions. In the case of the sphere, our approach provides a foundation for a new multiresolution approach to evaluating and estimating models of gravitational potentials used for satellite orbit computations. These results have been published as a part of the paper G. Beylkin and L. Monzon, "Approximation by exponential sums revisited", Applied and Computational Harmonic Analysis, v.28, pp. 131–149, 2010, (see **http://dx.doi.org/10.1016/j.acha.2009.08.011**).

These results served as a starting point for an STTR grant involving two companies, Numerica and Omitron. Also, local gravity models generated interest at NASA as an approach

for modeling gravity of irregular shaped bodies, e.g. asteroids. The work in this direction has already started.

(2) Current methods for solving ODEs, be that multistep or Runge-Kutta, are based on polynomial approximations of functions. However, both recent and classical results indicate that in many situations bandlimited functions, rather than polynomials, provide a near optimal tool for numerical integration and interpolation of functions. Given recently developed tools for computing with bandlimited functions, our goal has been to demonstrate that, by using bandlimited approximations, we gain advantages in numerical solution of the initial value problem for the ordinary differential equations (ODEs). Specifically, we considered ODEs for orbit determination as a practical application of our approach. Using the Gaussian-type quadratures for bandlimited functions, we developed ODE solvers that mimic the standard implicit Runge-Kutta methods with Gauss-Legendre nodes. Similar to such methods, our method is A-stable. Moreover, we show that in spite of the approximate nature of our quadratures (generated for a finite but arbitrary precision), the integrators may easily be made symplectic. The nodes of the quadratures do not concentrate excessively near the end points thus allowing us to compute large portions of an orbit at once. This, in turn, allows us to use initially a simplified gravity model (e.g., with only 3 terms) to approximate a large portion (e.g., 1/2 of a period) of an orbit by rapidly solving a system of nonlinear equations. We then access the full gravity model and evaluate the gravitational force at the nodes that, as a result of the initial low order approximation, are located fairly close to their correct positions. We then adjust the orbit without accessing the full gravity model. This results in an essentially correct trajectory. At this point we may (and currently do) access the full gravity model one more time to evaluate the gravitational force and perform another iteration. Thus, we access the full gravity model at most twice per node while using a number of nodes that is substantially lower than in traditional methods.
We have transferred a copy of the code to Jack M. Van Wieren, USAF AFSPC for testing and comparison with existing solvers. A draft paper is attached to this report.

(3) We have developed a new numerical approach for obtaining quadratures on the sphere that are invariant under the icosahedral group. These nearly optimal quadratures integrate all $(N+1)^2$ linearly independent functions in a rotationally invariant subspace of maximal order and degree N. The nodes of these quadratures are nearly uniformly distributed and the number of nodes is only marginally more than the optimal $(N+1)^2/3$ nodes. Using these quadratures, we discretize the reproducing kernel on a rotationally invariant subspace to construct an analogue of Lagrange interpolation on the sphere. This representation uses function values at the quadrature nodes. In addition, the representation yields an expansion that uses a single function centered and mostly concentrated at nodes of the quadrature, thus providing a much better localization than spherical harmonic expansions. We show that this representation may be localized even further. We also describe two algorithms of complexity $\mathcal{O}(N^3)$ for using these grids and representations. Finally, we note that our approach is also applicable to other discrete rotation groups. These results have

been published, see C. Ahrens and G. Beylkin, "Rotationally Invariant Quadratures for the Sphere", Proceeding of Royal Society of London A, vol. 465, pp.3103–3125, 2009 (see **http://dx.doi.org/10.1098/rspa.2009.0104**).
These quadratures form a foundation for the development of new multiresolution grids on the sphere with the goal of replacing spherical harmonics by localized representations that are more appropriate for gravity modeling and other tasks where physics requires local approximations. We have started work on the further localization of representations of functions on the sphere and developing multiresolution constructions suitable for estimation of functions on the sphere.

Below we provide background and motivations for these developments.

1. Local approximation of gravity models and a new type of ODE solvers

Space surveillance, navigation of aircraft and missiles requires detailed gravity and elevation maps. The choice of representation of such digital information affects efficiency and accuracy of numerical algorithms that use these data. It is often highly desirable to combine heterogeneous data derived from sources with different spatial resolutions, accuracies, and coverages. Thus, there is a need to develop data representations and algorithms that are flexible and efficient to satisfy all of these requirements.

Many of the current mathematical representations and associated numerical algorithms were developed in the 1950s and 1960s, at a time when computers were quite different from what is routinely available today. The difference is not only in the drastically improved speed of modern computers, but also in the size of available memory (RAM), the computer architecture (e.g., parallel architecture of inexpensive Graphics Processing Units (GPUs)), and the relative cost of arithmetic operations. These differences, as well as the availability of recently developed numerical methods, require rethinking of the traditional approach to many of these problems.

Spherical harmonic expansions have been and still are the preferred analytical tool for the representation of gravitational data. Among their advantages for low resolution models, is the relative simplicity, rapid convergence (although only for smooth functions), relatively low cost of evaluation at each point, relatively low memory requirements, and the availability of the Fast Fourier Transform (FFT) for the acceleration of the azimuthal part of the associated transform. As the gravity maps became more detailed, higher order spherical harmonic expansions had to be used to represent them. Currently, expansions of order and degree 360 are used in certain situations, and those of order and degree 720 are being constructed; there is every reason to expect that the spatial resolution of the maps will keep increasing. As a result of this growth, several problems with spherical harmonic expansions have become obvious, namely:

(1) the representation of data is global; a local change in the model in any one area requires re-calculation of the whole expansion.
(2) evaluating expansion of the order and degree n at a single point costs $\mathcal{O}(n^2)$ operations; in many situations this cost is excessive.

(3) typically the data are needed within a relatively small window, covering only a small part of the sphere; extracting these data from the spherical harmonic expansion is inconvenient and could be expensive.

Several local approximation of gravity models have been developed to address some of these issues. In this project we demonstrated that these approximations do not compromise accuracy in orbit determination but drastically improve the speed of these computations. On the other hand, we still do not have a fully adequate "replacement" for spherical harmonics and our effort has been advancing with this goal in mind.

A more detailed accuracy comparison of using local gravity models in orbit determination may be found in **http://dx.doi.org/10.2514/1.45336.** A new approach to estimating local gravity models has been suggested in **http://dx.doi.org/10.1016/j.acha.2009.08.011.**

Other aspects of these type of problems, namely, a new type of A-stable, symplectic integrator is describes in the attached preprint.

We also include below a brief description and motivation for constructing quadratures invariant under the icosahedral group and refer to **http://dx.doi.org/10.1098/rspa.2009.0104** for a more detailed description.

2. Nearly optimal quadratures on the sphere

Many problems in physics, mathematics and engineering involve integration and interpolation on the sphere in $\mathbb{R}^3$. Of particular importance are discretizations of rotationally invariant subspaces of $L^2(\mathbb{S}^2)$ that integrate all spherical harmonics up to a fixed order and degree. A typical approach to discretizing the sphere is that of equally spaced discretization in azimuthal angle and Gauss-Legendre discretization in polar angle, leading to an unreasonably dense concentration of nodes near the poles. It is well known that in a variety of applications such concentration of nodes may lead to problems when using these grids.

We develop a systematic numerical approach for constructing nearly optimal quadratures invariant under the icosahedral group to integrate rotationally invariant subspaces of $L^2(\mathbb{S}^2)$ up to a fixed order and degree. Using these grids and a reproducing kernel, we show how to replace the standard basis of spherical harmonics on a rotationally invariant subspace by a representation formed using a single function centered at the quadrature nodes. The reproducing kernel is mostly concentrated near the corresponding grid point. In the resulting representation, the coefficients, up to a factor, are the values on the grid of the function being represented. We may interpret this construction as an analogue of Lagrange interpolation on the sphere and note that it allows us to develop well conditioned linear systems for interpolation in contrast to some earlier constructions.

An alternative to spherical harmonics has long been sought, especially for numerical purposes. Spherical harmonics provide an efficient orthonormal basis, nicely subdivided into rotationally invariant subspaces. However, the global support of these functions poses a serious difficulty in problems where physical effects are localized. In fact, the global nature of spherical harmonics is a consequence of their optimality. Therefore, if we want localized functions to represent the same subspaces, we necessarily must have a less efficient representation.

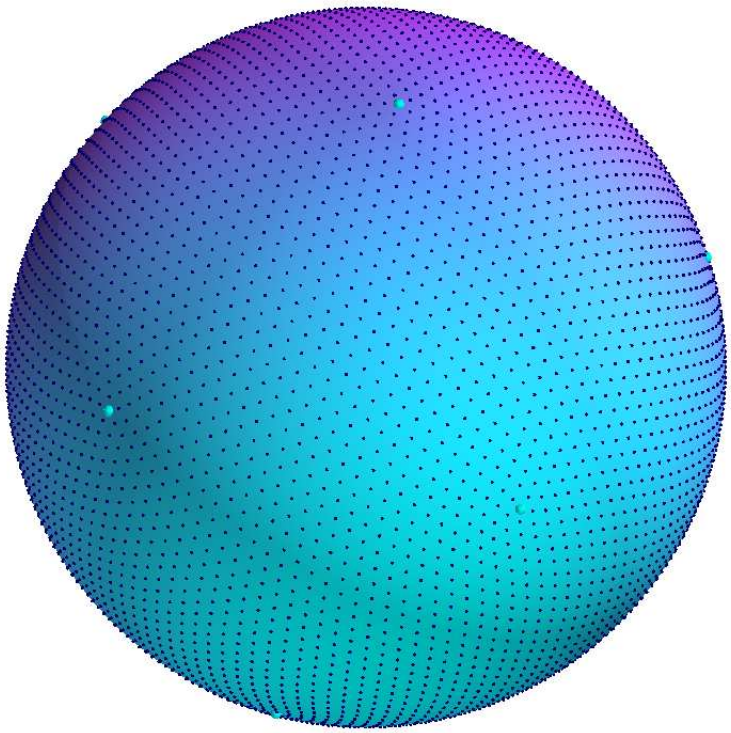

FIGURE 2.1. Positions of 7212 quadrature nodes of a quadrature integrating exactly all spherical harmonics in the subspace of maximal order and degree 145. This quadrature has efficiency $\eta = 0.98521....$ (efficiency$\eta = 1$ implies that each node accommodates 3 degrees of freedom in the subspace).

We view our approach as the first step in constructing a local and multiresolution representation of functions on the sphere that respects rotationally invariant subspaces. We note that the high efficiency of quadratures constructed in this paper implies a near uniform distribution of nodes on the sphere. On the other hand, the nodes maintain a regular organization visually similar to that of geodesic or equal area grids. Moreover, our grids are associated with rotationally invariant subspaces, an important property in a number of numerical applications, e.g., geodesy. To date, we have constructed grids which integrate subspaces of maximum order and degree ranging from 5 up to 210. As an example, we illustrate Figure 2.1 a grid with 7212 nodes integrating subspaces with maximal order and degree 145.

The rotationally invariant spherical grids constructed here have many applications. Let us mention a few specific problems in some detail. First, due to the central role played by spherical harmonics in the theory of gravity and magnetic fields, solutions to many geodetic problems use them as a basis. Yet, their global support is inconsistent with the physical nature of the problem

leading to many difficulties in, e.g., constructing gravity models. The grids developed in this paper provide a first step toward replacing spherical harmonics with localized functions. We plan to continue work in this direction. Second, the equations used in global atmospheric modeling are typically posed on the sphere. Current spectral methods which use spherical harmonics suffer from the above mentioned problems of nodal clustering and require additional steps to alleviate the problem. The new representations developed in this paper eliminate clustering and singularities due to the coordinate system and should provide efficient solution methods. Third, acoustic and electromagnetic scattering problems posed as integral equations involve integration over spherical domains. New algorithms for the numerical solution of these integral equations may be based on the results of this paper. Finally, we mention a numerical technique used in molecular dynamics calculations known as discrete variable representation (DVR). Our quadratures should extend such methods by allowing effectively an arbitrary order and degree.

Current address: Department of Applied Mathematics, University of Colorado at Boulder, 526 UCB Boulder, CO 80309-0526,

ODE Solvers Using Bandlimited Approximations

G. Beylkin and K. Sandberg

Department of Applied Mathematics
University of Colorado at Boulder
526 UCB
Boulder, CO 80309-0526

Abstract

We use generalized Gaussian quadratures for exponentials to develop a new ODE solver. These generalized Gaussian quadratures integrate functions e^{ibx} for all $|b| \leq c$, where the nodes and weights are computed for a given bandlimit c and any user selected accuracy ϵ. An improrant property of nodes of these quadratures is that they do not concentrate excessively near the end points of the interval as the nodes of the standard polynomial-based Gaussian quadratures. The new ODE solver is analogous to the usual implicit Runge Kutta (collocation) method but it allows us to use a large number of nodes due to properties of the quadrature. We show that the resulting ODE solver is symplectic and A-stable. We use this solver in the problem of orbit determination and achieve speed close to that of an explicit multistep method.

1 Introduction

Current methods for solving ODEs, be that multistep or Runge-Kutta, are based on polynomial approximations of functions. On the other hand, both recent and classical results [5,4,18,16,11,17] indicate that in many situations bandlimited functions provide a near optimal tool for numerical integration and interpolation of functions. For example, a time domain solver for the wave equation [5] uses bandlimited approximations and yields about 12 digits of accuracy with only 3 nodes per wavelength. Given recently developed tools for computing with bandlimited functions, our goal is to demonstrate that, by using bandlimited approximations, we also gain advantages in numerical solution

[1] This research was partially supported by AFOSR grant FA9550-07-1-0135, NSF grant DMS-0612358, DOE/ORNL grants 4000038129 and DE-FG02-03ER25583.

of the initial value problem for the ordinary differential equations (ODEs). As an example, we use ODEs for orbit determination to provide us with both, a practical application as well as a gauge to ascertain the performance of new algorithms.

Numerical solution of ODEs is a mature area of applied mathematics with many well-developed software packages. In spite of this satisfactory state of affairs, we challenge the usual approach to selecting an ODE solver for a given problem. We note a specific inefficiency of multistep methods, the fact that such methods can only be A-stable if their order does not exceed 2 (the so-called Dahlquist barrier). Whereas implicit Runge-Kutta methods with Gauss-Legendre nodes are A-stable and symplectic, we are limited in using a large number of Gauss-Legendre nodes per time step since such nodes concentrate excessively near the end points of the time interval. We note that unlike in problems of wave propagation [5] where solutions are naturally expressed via exponentials, solutions of some ODEs may, in fact, be polynomials or other functions that do not have an efficient approximation via exponentials. For such equations it actually may be advantageous to use polynomial quadratures. However, it is more typical to encounter ODEs where solutions do have an efficient approximation via exponentials and our method is most suitable for such ODEs.

Unlike the classical Gaussian quadratures for polynomials, the Gaussian type quadratures for exponentials attempt to integrate an infinite set of functions, namely, e^{ibx} with $|b| \leq c$, using a finite set of nodes. Clearly, there is no way to accomplish this exactly. Thus, these quadratures are constructed so that all exponentials for $|b| \leq c$ are integrated with accuracy of at least ϵ, where ϵ is arbitrarily small but finite. Such quadratures were constructed in [4,18] (we use those in [4]). An important observation in [5] is that the nodes of quadratures of this type do not concentrate excessively near the end points, thus allowing us to use as many nodes as necessary, without a penalty due to their number. The density of nodes increases toward the end points of the interval only by a factor that depends on the desired accuracy but not on the overall number of nodes. Since we may choose many nodes, say M, it makes sense to ask if the integration matrix of an implicit Runge-Kutta type method can be applied in $\mathcal{O}(M)$ or $\mathcal{O}(M \log M$ operations rather $\mathcal{O}(M^2)$. We note that such question does not make sense for the classical, polynomial-based quadratures since a rapid concentration of nodes towards the end points severely limits the number of nodes. We show that, indeed, it is possible to apply the integration matrix in a fast manner (within a finite accuracy ϵ). This may, for example, be accomplished by using the partitioned low rank (PLR) representation as it was described in [BEY-SAN:2005]. Using PLR representation, the speed of an implicit method is now of the same order as that of an explicit method. We note that there might be even more efficient approaches to the fast application of the integration matrix than PLR representation.

This change of complexity allows us to challenge the usual view that explicit methods are inherently faster than implicit methods. While explicit methods require a significant oversampling, implicit methods only needs to maintain minimally appropriate sampling but do extra work to solve a system of equations at each step. If the efficiency is measured in terms of the number of function calls, then their total number is no longer differs substantially between explicit and implicit methods.

Using the Gaussian-type quadratures for bandlimited functions, we construct ODE solvers that mimic the standard implicit Runge-Kutta methods with Gauss-Legendre nodes (see e.g. [ISERLE:1996]). Similar to such methods, our method is A-stable. Although our algorithms are based on approximate quadratures and, thus, cannot produce results with accuracy better than the chosen accuracy ϵ, we note that if $\epsilon \approx 10^{-16}$ the effect of such limitation is the same as using double precision in floating point arithmetic. Moreover, we show that in spite of the approximate nature of our quadratures, the integrators may be made exactly symplectic.

All these properties make our approach attractive for a number of applications and we select orbit determination as an example. Not only does it provide us with a practical application, it also allows us to demonstrate certain "tricks" associated with our choice of the method that further reduce the computational cost. Although orbit determination typically is not a stiff problem, we demonstrate advantages of using an implicit method in that we substantially reduce the required number of function calls to the full gravity model (beyond the reduction of overall number of function calls in comparison with existing methods).

2 Preliminaries: quadratures for bandlimited functions

2.1 Bandlimited functions as a replacement of polynomials

Recently constructed quadratures [18,4] address efficiency of sampling for representing functions by breaking with the conventional approach of using polynomials as the fundamental tool in analysis and computation. The approach based on polynomial approximations has a long tradition and leads to such notions as the order of convergence of numerical schemes, Gaussian quadratures (for polynomials), polynomial based interpolation, and so on. Recently, the dominance of such an approach (it clearly remains reasonable for many problems) has been successfully challenged. It turns out that constructing quadratures for bandlimited functions, e.g., exponentials e^{ibx}, with $|b| \leq c$, where c is the bandlimit, a fixed parameter, leads to significant improvement

in performance of algorithms for interpolation, estimation and solving partial differential equations [5].

2.2 Bases for bandlimited functions

Whenever measurements are performed, all sensors are of limited size; their frequency response for all practical purposes is also limited outside a finite range. On the other hand, it is well-known that a function whose Fourier Transform has compact support can not have compact support itself (unless it is identically zero). Therefore, it is natural to analyze an operator whose effect on a function is to truncate it both in the original and the Fourier domains. This has been the topic of a series of seminal papers by Slepian et al. [17], [11], [12], [14], [15], where it is observed (*inter alia*) that the eigenfunctions of such operator (see (1) below) are the Prolate Spheroidal Wave Functions (PSWFs) of classical Mathematical Physics.

While *periodic* bandlimited functions may be expanded into Fourier series, and band-limited functions on the *real line* may be represented via the Fourier Integral Transform, we must also deal with non-periodic functions on *intervals*, where neither the Fourier series nor the Fourier Integral can be used efficiently. This motivates the introduction of a basis that efficiently represents (on an interval) functions of the form e^{ibx} for an arbitrary real value b, as long as $|b| < c$, with a fixed (bandlimit) parameter c. Since b varies continuously, such a basis is not finite. On the other hand, an arbitrary but finite precision is achievable with a finite basis consisting, for example, of appropriately chosen PSWFs.

For a real number $c > 0$ (to be referred to as the bandlimit), we consider the operator $F_c : \ L^2[-1,1] \to L^2[-1,1]$, defined by the formula

$$F_c(\psi)(\omega) = \int_{-1}^{1} e^{icx\omega}\psi(x)dx, \tag{1}$$

and the operator $Q_c = \frac{c}{2\pi}F_c^* F_c$; it is easily seen that

$$Q_c(\psi)(y) = \frac{1}{\pi}\int_{-1}^{1} \frac{\sin(c(y-x))}{y-x}\psi(x)\ dx. \tag{2}$$

The eigenfunctions $\psi_0^c, \psi_1^c, \psi_2^c, \cdots$ of Q_c coincide with those of F_c, and the eigenvalues μ_j of Q_c are connected with the eigenvalues λ_j of F_c by the formula

$$\mu_j = \frac{c}{2\pi}|\lambda_j|^2, \tag{3}$$

for all $j = 0, 1, 2, \cdots$, where the eigenvalues are ordered by decaying magnitude.

In many respects, PSWFs are strikingly similar to orthogonal polynomials; they are orthonormal, constitute a Chebychev system, and admit a version of Gaussian quadratures (see [18], [4]). One of the results in [18] and [4] is formulated as

Proposition: *For $c > 0$ and $\epsilon > 0$, there exist nodes $-1 < \theta_1 < \theta_2 < \cdots < \theta_M < 1$ and corresponding weights $w_k > 0$, such that for any $x \in [-1, 1]$,*

$$\left| \int_{-1}^{1} e^{ictx} \, dt - \sum_{k=1}^{M} w_k e^{ic\theta_k x} \right| < \epsilon, \tag{4}$$

where the number of nodes, M, is (nearly) optimal. The nodes and weights maintain the natural symmetry, $\theta_k = -\theta_{M-k+1}$ and $w_k = w_{M-k+1}$. When the functions $\psi_0^c, \psi_1^c, \psi_2^c, \cdots, \psi_{M-1}^c$ are used as a basis for the interpolation on the interval $[-1, 1]$, with the points $\theta_1, \theta_2, \cdots, \theta_M$ used as the interpolation nodes, the resulting interpolation formula is stable.

This proposition provides a tool for the numerical integration and interpolation of functions of the form e^{ibx} on $[-1, 1]$. The nodes and weights are functions of both the bandlimit $c > 0$ and the accuracy $\epsilon > 0$, and may be viewed as the generalized Gaussian quadratures for the bandlimited functions. It is worth noting that the algorithm in [4] identifies the nodes of the generalized Gaussian quadratures in (4) as zeros of the *discrete* prolate spheroidal wave functions (DPSWFs) corresponding to small eigenvalues.

One of problems associated with the numerical use of orthogonal polynomials, is the concentration of their roots near the ends of the interval of their definition. Using nodes $\theta_1, \theta_2, \cdots, \theta_M$, we maintain a nearly optimal sampling rate, close to the Nyquist rate for periodic functions. Let us consider the ratio

$$r(M, \epsilon) = \frac{\theta_2 - \theta_1}{\theta_{\lfloor M/2 \rfloor} - \theta_{\lfloor M/2 \rfloor - 1}}, \tag{5}$$

where "$\lfloor M/2 \rfloor$" denotes the least integer part. Observing that the distance between nodes of Gaussian quadratures changes monotonically from the middle of an interval toward its end points, and that the smallest distance occurs between the nodes closest to the end point, the ratio 5 may be used as a measure of node accumulation. For example, the distance between the nodes near the end points of the standard Gaussian quadratures for polynomials decreases as $\mathcal{O}(1/M^2)$, so that we have $r(M, \epsilon) = \mathcal{O}(1/M)$, where M is the number of nodes,

In Figure 1, considering bandlimit c as a function of the number of nodes, M, and the desired accuracy ϵ, we observe that the oversampling factor,

$$\alpha(M, \epsilon) = \frac{\pi M}{c(M, \epsilon)} > 1,$$

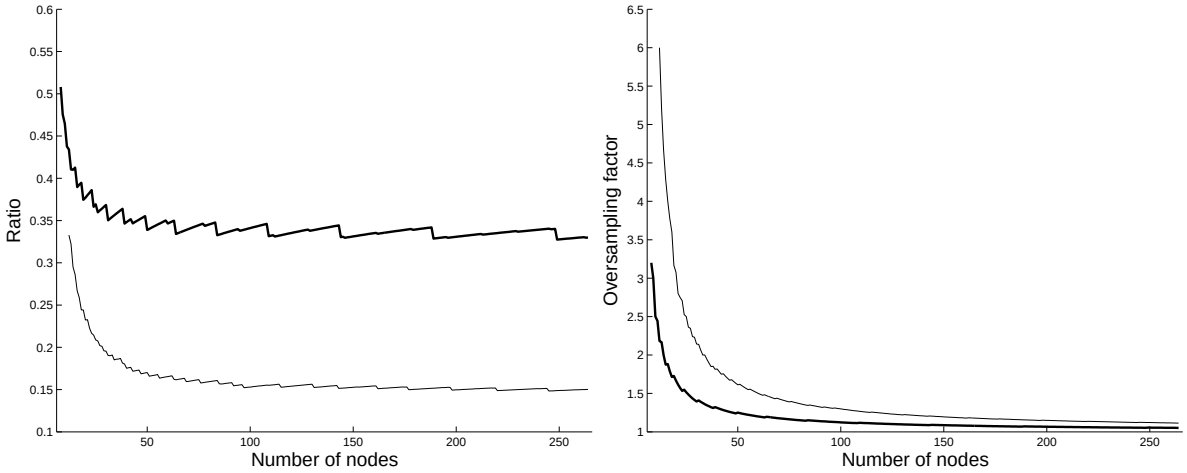

Figure 1. The ratio $r(M, \epsilon)$ in (5) and the oversampling factor $\alpha(M, \epsilon)$ plotted against the number of nodes for quadratures of accuracy $\epsilon \approx 10^{-7}$and $\epsilon \approx 10^{-17}$.

approaches 1 for large M. This factor compares the critical rate of sampling of a smooth *periodic* function, either for integration or interpolation, to that of smooth *non-periodic* function defined on an interval. We recall that in the case of the Gaussian quadratures for polynomials, the oversampling factor approaches $\frac{\pi}{2}$ rather than 1 (see e.g. [8]).

To illustrate the gain in efficiency, we compare accuracy of differentiation using bases for bandlimited functions, finite differences and spectral differentiation using the Chebyshev polynomials. We construct two derivative matrices using bandlimited bases with accuracy $\epsilon = 10^{-7}$ and bandlimit $c = 23\pi$, and with $\epsilon = 10^{-13}$ and bandlimit $c = 18.5\pi$. For comparison, we construct a second-order central finite-difference derivative matrix and, for spectral differentiation, a block diagonal spectral derivative matrix with 4 blocks where each diagonal block is a derivative matrix with respect to the first 16 Chebyshev polynomials. We differentiate the function $f(x) = \sin(bx)$ for 200 values of b, $-32\pi \leq b \leq 32\pi$. In Figure 2 we illustrate the extra bandwidth where the bandlimited derivatives are accurate [5].

3 Discretization of Picard integral equation

Consider the initial value problem

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \ \mathbf{y}(0) = \mathbf{y_0},$$

or, equivalently,

$$\mathbf{y}(t) = \mathbf{y_0} + \int_0^t \mathbf{f}(s, \mathbf{y}(s)) \ ds. \tag{6}$$

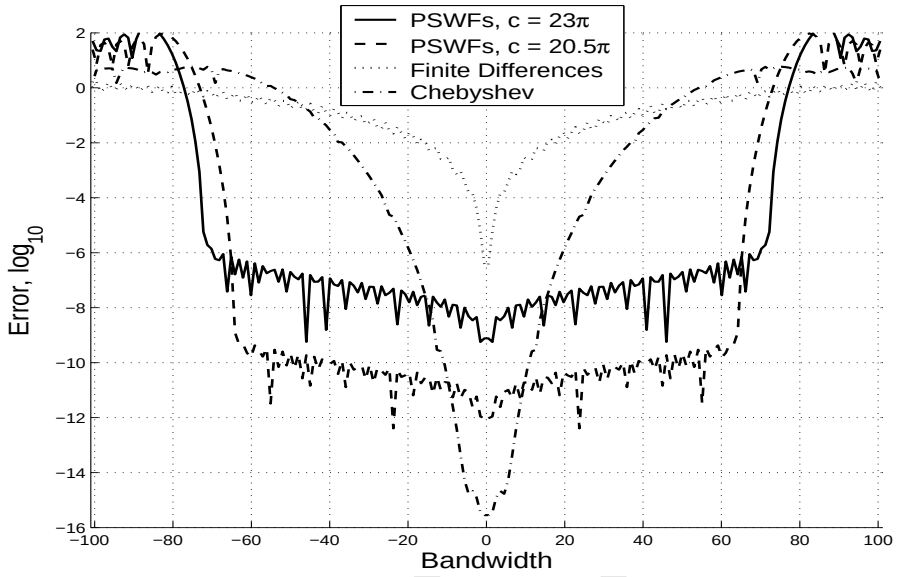

Figure 2. Comparison of absolute errors for the first derivative of the function $\sin(bx)$ in the interval $[-1, 1]$ with $|b| \leq 32\pi$. In all examples the derivative matrices use 64 independent parameters.

For a given accuracy ϵ, we select Gaussian nodes $\{\tau_j\}_{j=1}^{M}$ such that on $[0, t]$

$$\|\mathbf{f}(t, \mathbf{y}(t)) - \sum_{j=1}^{M} \mathbf{f}(\tau_j, \mathbf{y}(\tau_j)) R_j(t)\| \leq \epsilon, \tag{7}$$

where $R_j(t)$ are interpolating basis functions associated with the generalized Gaussian nodes for exponentials [4,5]. We discretize (6) by using (7) to obtain a nonlinear system,

$$\mathbf{y}(\tau_i) = \mathbf{y_0} + \sum_{j=1}^{M} \mathbf{f}(\tau_j, \mathbf{y}(\tau_j)) \int_0^{\tau_i} R_j(s) ds = \mathbf{y_0} + \sum_{j=1}^{M} S_{ij} f(\tau_j, \mathbf{y}(\tau_j)), \tag{8}$$

where $S_{ij} = \int_0^{\tau_i} R_j(s) ds$ is the integration matrix.

3.1 Relation to Implicit Runge Kutta methods based on collocation

Consider (8) with M quadrature nodes $\{\tau_j\}_{j=1}^{M}$ (and the corresponding weights $\{w_j\}_{j=1}^{M}$) in the interior of the interval $[0, t]$. Then implicit Runge-Kutta methods based on collocation discretize (6) as

$$\mathbf{y}(t) = \mathbf{y_0} + \sum_{j=1}^{M} w_j \mathbf{f}(\tau_j, \mathbf{y}(\tau_j)), \tag{9}$$

where $\{\mathbf{y}(\tau_j)\}_{j=1}^{M}$ are obtained by solving the nonlinear system (8). The nodes, weights, and the entries of the integration matrix are typically organized in

the Butcher tableau,

$$\begin{array}{c|c} \tau & S \\ \hline & w^t \end{array}.$$

Unlike in the standard implicit Runge-Kutta method based on Gauss-Legendre quadratures, we solve (8) on a time interval containing a large number of quadrature nodes, M (since these nodes do not concentrate excessively near the end points).

3.2 Exact Linear Part

In many problems (including that of orbit determination), the right hand side of the ODE, $\mathbf{f}(t, \mathbf{y})$, may be split into a linear and nonlinear parts,

$$\mathbf{f}(t, \mathbf{y}(t)) = \mathbf{L}\mathbf{y}(t) + \mathbf{g}(t, \mathbf{y}(t))., $$

so that the integral equation (6) may be written as

$$\mathbf{y}(t) = e^{tL}\mathbf{y_0} + \int_0^t e^{(t-s)\mathbf{L}}\mathbf{g}(s, \mathbf{y}(s))\, ds. \tag{10}$$

If the operator $e^{t\mathbf{L}}$ is computed efficiently, this formulation may lead to savings when solving the integral equation iteratively. We discretize (10) by using (7) and obtain

$$\begin{aligned} \mathbf{y}(\tau_i) &= e^{\tau_i\mathbf{L}}\mathbf{y_0} + \sum_{j=1}^{M} e^{(\tau_i-\tau_j)\mathbf{L}}\mathbf{g}(\tau_j, \mathbf{y}(\tau_j)) \int_0^{\tau_i} R_j(s)ds \\ &= e^{\tau_i\mathbf{L}}\mathbf{y_0} + \sum_{j=1}^{M} S_{ij} e^{(\tau_i-\tau_j)\mathbf{L}}\mathbf{g}(\tau_j, \mathbf{y}(\tau_j)) \end{aligned} \tag{11}$$

where $S_{ij} = \int_0^{\tau_i} R_j(s)ds$. We note that (8) is a special case of (11) with $\mathbf{L} = 0$ and $\mathbf{g} = \mathbf{f}$.

3.3 Algorithm

Next we describe a fixed point iteration to solve (11). Let N_{it} denote the number of iterations, which can either be set to a fixed number or determined adaptively. Labeling the intermediate solutions in the iteration scheme as $\mathbf{y}^{(i)},\ i = 1, \ldots, N_{it}$, we have

(1) Initialize $y^{(1)}(\tau_i) = \mathbf{y_0},\ i = 1, \ldots, M$.
(2) **For** $k = 1, \ldots, N_{it}$
For $m = 1, \ldots, M$

(a) Update the solution at the node m:

$$\mathbf{y}^{(k)}(\tau_m) = e^{\tau_m \mathbf{L}} \mathbf{y_0} + \sum_{j=1}^{M} S_{mj}\, e^{(\tau_m - \tau_j)\mathbf{L}} \mathbf{g}(\tau_j, \mathbf{y}^k(\tau_j))$$

(b) Update the right hand side: $\mathbf{g}(\tau_m, y^{(k)}(\tau_m))$

4 Symplectic integrators

Following [13], let us introduce matrix $M = \{m_{kj}\}_{k,j=1}^{M}$ for an implicit M-stage Runge-Kutta (RK) scheme,

$$m_{kj} = w_k S_{kj} + w_j S_{jk} - w_k w_j, \tag{12}$$

where $w = \{w_k\}_{k=1}^{M}$ and $S = \{S_{kj}\}_{k,j=1}^{M}$ define the Butcher's tableau for the method.

It is shown in [13] that

Theorem 1 *If matrix $M = 0$ in (12), then an implicit M-stage RK scheme is symplectic.*

This condition, $M = 0$, is satisfied for the Gauss-Legendre RK methods, see e.g. [6,13]. We will enforce it to construct symplectic integrators for collocation-type methods based on the generalized Gaussian quadratures for bandlimited functions. In spite of the fact that such quadratures are accurate only up to a fixed (but arbitrary) accuracy ϵ, we can still satisfy the condition $M = 0$ exactly.

Our proof is based on several observations.

Proposition 2 *Let $\{R_i(\tau)\}_{i=1}^{M}$ be interpolating basis functions for the quadrature nodes $\{\tau_j\}_{j=1}^{M}$, such that $R_i(\tau_j) = \delta_{ij}$. Define $F_i(\tau) = \int_{-1}^{\tau} R_i(s)\, ds$ and let $\{w_j\}_{j=1}^{M}$ be quadrature nodes such that*

$$\left| \int_{-1}^{1} F_j(\tau) F_i'(\tau)\, d\tau - \sum_{k=1}^{M} w_k F_j(\tau_k) F_i'(\tau_k) \right| < \epsilon^2 \tag{13}$$

Then

$$\left| \int_{-1}^{\tau_i} R_j(s)\ ds - \frac{\int_{-1}^{1} \int_{-1}^{\tau} R_j(s)\ ds\ R_i(\tau)\ d\tau}{w_i} \right| < \epsilon^2.$$

PROOF. We first observe that due to the interpolating property of $R_i(\tau)$, we have

$$\int_{-1}^{\theta_i} R_j(s)\ ds = \frac{\int_{-1}^{\theta_i} R_j(s)\ ds\ w_i R_i(\theta_i)}{w_i} = \frac{\sum_{k=1}^{M} \int_{-1}^{\theta_k} R_j(s)\ ds\ w_k R_i(\theta_k)}{w_i}. \quad (14)$$

Next we note that

$$\int_{-1}^{1} \int_{-1}^{\tau} R_j(s)\ ds\ R_i(\tau)\ d\tau = \int_{-1}^{1} F_j(\tau) F_i'(\tau)\ d\tau,$$

so that proposition follows by combining (14) and (13).

Proposition 3 *Let $\{R_i(\tau)\}_{i=1}^{M}$ be interpolating basis functions for the quadrature nodes $\{\tau_j\}_{j=1}^{M}$, such that $R_i(\tau_j) = \delta_{ij}$. Let $\{w_j\}_{j=1}^{M}$ be the corresponding quadrature weights so that*

$$\left| \int_{-1}^{1} R_i(\tau)\ d\tau - \sum_{k=1}^{M} w_k R_i(\tau_k) \right| < \epsilon^2.$$

Then

$$\left| \int_{-1}^{1} R_i(s)\ ds - w_i \right| < \epsilon^2.$$

PROOF. The result follows from the interpolating property of $R_i(\tau)$.

Theorem 4 *Given M quadrature nodes $\{\tau_k\}_{k=1}^{M}$ and interpolating functions $\{R_k\}_{k=1}^{M}$, let weights for the quadrature be defined as*

$$w_k = \int_{-1}^{1} R_k(\tau) d\tau \quad (15)$$

and the integration matrix as

$$S_{kl} = \frac{\int_{-1}^{1} \left(\int_{-1}^{\tau} R_l(s)\ ds \right)\ R_k(\tau)\ d\tau}{w_k}, \quad k, l = 1, \ldots, M.$$

Then

$$w_k S_{kl} + w_l S_{lk} - w_k w_l = 0,$$

and the implicit scheme using these nodes and weights is symplectic.

PROOF. Using propositions above, we observe that the weights defined in (15) are the same (up to accuracy ϵ^2) as those of the quadrature. The result follows if we set $F_k(\tau) = \int_{-1}^{\tau} R_k(\tau)\ d\tau$, $F_k'(\tau) = R_k(\tau)$ and integrate by parts to obtain

$$\begin{aligned} w_k S_{kl} + w_l S_{lk} - w_k w_l &= \int_{-1}^{1} F_l(\tau) F_k'(\tau)\ d\tau + \int_{-1}^{1} F_k(\tau) F_l'(\tau)\ d\tau - w_k w_l \\ &= F_l(1) F_k(1) - w_k w_l. \end{aligned}$$

By the definition of the weights we have $F_k(1) = w_k$ and, hence, $F_l(1)F_k(1) - w_k w_l = 0$.

Remark 5 *Implicit Runge-Kutta methods based on Gauss-Legendre nodes are A-stable (see e.g [10]). We verified numerically (within the accuracy of underlying quadrature) that this property carries on to the schemes based on the generalized Gaussian quadratures for exponentials. Currently, we do not have a proof of this observation.*

5 Fast application of integration matrix

The repeated application of the integration matrix S dominates the computational cost of current algorithm since applying S as a dense matrix requires $\mathcal{O}(M^2)$ operations. We demonstrate that this cost may be reduced to $\mathcal{O}(M \log M)$. However, we consider this as a preliminary result since it is also important to assure that the break even point with the usual dense matrix-vector multiply is low and the fast scheme becomes beneficial for a relatively small M. We note that the overall operation count also takes into account the number of iterations N_{it} for solving the nonlinear system (11) in Section˜3.3. Since the number of iterations depends on the length of the time interval on which we discretize our system of ODEs, it may grow as a function of M and diminish the benefit of increasing the number of nodes M.

We now outline two approaches for applying the integration matrix S in $O(M \log M)$ operations.

5.1 The Partitioned Low Rank (PLR) representation

The Partitioned Low Rank (PLR) representation may be applied in a variety of problems where the off-diagonal part of a matrix has a relatively small rank. In particular, PLR may be used for integration and differentiation matrices. The idea is to subdivide the matrix as in Figure 3 and then represent individual

off-diagonal blocks as a sum of rank one matrices,

$$\sum_{k=1}^{r} \mathbf{u}_k \mathbf{v}_k,$$

where $\mathbf{u}_k$ and $\mathbf{v}_k$, $k = 1, \ldots, r$ are vectors of appropriate size for a given block. In this representation, the number of terms r (the rank of the off-diagonal block) is selected for a given user-supplied accuracy and may be found by the Singular Value Decomposition (SVD). However, since we do not require orthogonality between vectors, a simpler algorithm may be used instead. If the ranks of off-diagonal blocks r are small, the cost of applying matrces in the PLR representation is $O(M \log M)$. For more details, see [2,5].

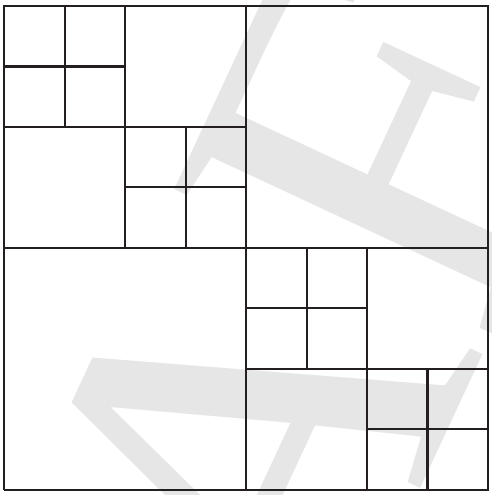

Figure 3. Partitioning of a matrix in PLR representation

As an example, consider the integration matrix in Theorem 4 with 70 nodes, a size that we have found to give a good compromise between large number of nodes vs. small number of iterations. In this case, optimal PLR performance is obtained by using just one level of subdivision with the rank $r = 13$ for the lower and $r = 14$ for the upper off-diagonal blocks to attain double precision accuracy. In this case, applying the integration matrix in the PLR representation requires 7735 operations, whereas applying the dense integration matrix takes 9800 operations, only a moderate improvement in performance.

It remains a challenge to increase the intervals of integration (in order to work with a larger matrix size M) without increasing the number of iterations.

5.2 Quadratures of Gauss-Trapezoidal type

In [1] Alpert introduced high order, polynomial based quadratures with nodes equally spaced in the interior of the interval and with a limited region of

higher node density near its boundary. Such node distribution allows one to use Fourier-type methods combined with a low rank correction. It turns out that a similar quadrature for exponentials (bandlimited functions) may also be constructed and we use an example of such quadratures (supplied to us by Brad Alpert) in our numerical experiments. We note that this arrangement of nodes does reduce the bandlimit of functions accurately integrated by the quadrature but its structure provides certain advantages vis-a-vie fast algorithms.

Let us demonstrate how this quadrature may be used for fast application of the integration matrix defined by

$$S_{kl} = \int_{-1}^{\theta_k} R_l(s)\ ds.$$

We note that although currently we do not have a proof that this matrix is symplectic, we verified numerically that it is very close to the symplectic integration matrix defined in Theorem 4.

Using representation of the interpolating functions $R_l(s)$ via exponentials (i.e., their definition in [5]), we have

$$R_l(s) = \sum_{j=1}^{M} A_{lj} e^{ic\theta_j s} \tag{16}$$

and write S_{kl} as

$$S_{kl} = \tilde{S}_{kl} - \alpha_l$$

where

$$\tilde{S}_{kl} = \sum_{j=1}^{M} A_{lj} \frac{e^{ic\theta_k\theta_j}}{ic\theta_j}$$

and

$$\alpha_l = \sum_{j=1}^{M} A_{lj} \frac{e^{-ic\theta_j}}{ic\theta_j}.$$

We note that matrix elements α_l are independent of the index k and, therefore, may be applied as a rank-one matrix. Hence, we only need to consider applying $\tilde{S}$. Let us define

$$E_{kj} = e^{ic\theta_k\theta_l}$$

and the diagonal matrix

$$D_{kj} = \begin{cases} \frac{1}{ic\theta_j},\ i = j \\ 0 \quad\ i \neq j \end{cases}.$$

Since functions $R_l(s)$ are interpolating, $R_l(\theta_k) = \delta_{kl}$, and we have from (16) that $A = E^{-1}$. Hence, we write

$$\tilde{S} = E^{-1} D E.$$

Let us introduce matrix of the disrete Fourier Transform,

$$F_{kj} = e^{2\pi ikj/M},$$

and a parameter

$$c = \frac{L^2\pi}{2M},$$

where $L = 2a + n_{interior} - 1$, $n_{interior}$ is the number of interior equispaced nodes and a is the size of the region near the boundary with nodes of higher density. We then have for the interior nodes

$$E_{kj} = e^{ic\theta_k\theta_l} = e^{ic(2k/L-1)(2j/L-1)} = e^{2\pi ikj/M}e^{-2ick/L}e^{-2icj/L}e^{ic}$$

or

$$E_{kj} = F_{kj}e^{-2ick/L}e^{-2icj/L}e^{ic},$$

where

$$D^F = \mathrm{diag}\{e^{-2icj/L}e^{ic/2}\}_{k=0}^{M-1}$$

For the full matrix E we write

$$E = D^F F D^F + G$$

where G is a matrix that with zeros in its interior and a non-zero border near the edges of the matrix. Thus, matrix G has low-rank.

Applying matrix S to a vector f may now be written as

$$S\mathbf{f} = (D^F F D^F + G)^{-1}\Big(D(D^F F D^F + G)\mathbf{f}\Big) = (D^F F D^F + G)^{-1}\mathbf{g},$$

where

$$\mathbf{g} \equiv \Big(D(D^F F D^F + G)\mathbf{f}\Big)$$

We note F is a matrix of the Discrete Fourier Transform (DFT) which may be applied in $O(M \log M)$ operations using the FFT. Since G is of low rank and D and D^F are diagonal, $\mathbf{g}$ may be computed in $O(M \log M)$ operations. Furthermore, we have

$$(D^F F D^F + G)^{-1} = (I + D^{F,-1}F^{-1}D^{F,-1}G)^{-1}F^{-1}$$

and, using the fact that $D^{F,-1}F^{-1}D^{F,-1}G$ is of low rank, we have a fast algorithm to compute $S\mathbf{f}$ by the use low rank update of the inverse.

6 Numerical Examples

6.1 Orbit Determination

Current ODE integrators for orbit computations use high order Gauss-Jackson multistep methods. Several years ago we have compared the number of function evaluations of this method with an explicit version of the deferred spectral correction method in [7,9]. The comparison was in favor of the latter and it showed that high order multistep methods are significantly oversampled (which also follows from theoretical considerations). Therefore, a possible gain in performance should come from the reduced sampling requirements; we believe that quadratures for bandlimited functions are the right tool for this purpose. The new methods will add to the variety of available techniques and will also provide new time evolution schemes for partial differential equations.

Let us consider the spherical harmonic model of the gravitational potential, $V(x, y, z)$, which in the spherical system of coordinates is written in terms of the spherical harmonics,

$$V_{sph}(\rho, \phi, \theta) = \frac{\mu}{a\rho}\left(1 + \sum_{n=2}^{N} \rho^{-n} Y_n(\theta, \phi)\right). \tag{17}$$

Let $\mathbf{G} = (G_x, G_y, G_z)^t = \nabla V$ and we use the gravity model of the form

$$\mathbf{G}(t, \mathbf{y}(t)) = \begin{bmatrix} -\frac{\mu x(t)}{(x(t)^2+y(t)^2+z(t)^2)^{3/2}} + \sum_{n=2}^{N} G_x(\mathbf{y}(t)) \\ -\frac{\mu y(t)}{(x(t)^2+y(t)^2+z(t)^2)^{3/2}} + \sum_{n=2}^{N} G_y(\mathbf{y}(t)) \\ -\frac{\mu z(t)}{(x(t)^2+y(t)^2+z(t)^2)^{3/2}} + \sum_{n=2}^{N} G_z(\mathbf{y}(t)) \end{bmatrix}.$$

In order to reduce the number of function calls to the full model, we initially use the gravity model with $N = 3$ on a large portion (e.g., 1/2 of a period) of an orbit to solve the system of nonlinear equations via a fixed point iteration. We then access the full gravity model and evaluate the gravitational force at the nodes that by now are located close to their correct positions. We continue iteration (without accessing the full gravity model again) to adjust the orbit. This results in an essentially correct trajectory. At this point we may (and currently do) access the full gravity model one more time to evaluate the gravitational force and perform another iteration. Thus, we access the full gravity model at most twice per node while using a number of nodes that is substantially lower than in traditional methods.

Let us now re-write the orbit determination problem (using only gravitational forces) in a form that conforms with the algorithm in Section 3.3. We define

the six component vector

$$\mathbf{y}(t) = \begin{bmatrix} x(t) \\ x'(t) \\ y(t) \\ y'(t) \\ z(t) \\ z'(t) \end{bmatrix},$$

where $[x, y, z]^t$ denote the positions, and $[x', y', z']^t$ denote the velocities. We also define the matrix

$$\mathbf{L} = \begin{bmatrix} 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0 \end{bmatrix}$$

and the right hand side

$$\mathbf{g}(t, \mathbf{y}(t)) = \begin{bmatrix} 0 \\ -\frac{\mu x(t)}{(x(t)^2+y(t)^2+z(t)^2)^{3/2}} + \sum_{n=2}^{N} G_x(\mathbf{y}(t)) \\ 0 \\ -\frac{\mu y(t)}{(x(t)^2+y(t)^2+z(t)^2)^{3/2}} + \sum_{n=2}^{N} G_y(\mathbf{y}(t)) \\ 0 \\ -\frac{\mu z(t)}{(x(t)^2+y(t)^2+z(t)^2)^{3/2}} + \sum_{n=2}^{N} G_z(\mathbf{y}(t)) \end{bmatrix},$$

where $\mathbf{G}(\mathbf{y}(t)) = [\mathbf{G}_x(\mathbf{y}(t)), \mathbf{G}_y(\mathbf{y}(t)), \mathbf{G}_z(\mathbf{y}(t))]$ denotes the x, y and z-components of gravity model starting with $n = 2$ and up to order N. The ODE describing the orbit determination problem is now given by (10).

We apply the algorithm in Section 3.3 by first using $N_{it} = 2$ and $N = 3$ to obtain an approximate solution. We then switch to the full model with $N = 70$ and, at this point require only one or two iterations N_{it}. Thus, we need to access the full model only once or twice per node.

6.2 Timing

We tested and timed the algorithm by computing an orbit of 86000 seconds ($\approx$ 1 day) using a 36 and 70 degree gravitational model. The timing was performed on a computer with an Intel Core2 Extreme processor at 2.96 GHz and 4 GB of RAM at 800 MHz. (However, only a small fraction of the RAM was actually used during the computation.)

6.2.1 Timing results using spherical harmonics

We first timed the algorithm using the spherical harmonic gravitational model. The timing results are given in Table 1.

Degree	CPU Time (s)	Number of function calls
36	9.0E-02	6160
70	2.7E-01	6160

Table 1
CPU time and number of function calls to full gravitational model when computing a orbit of 86000 seconds using spherical harmonics.

6.2.2 Timing results using a cubed sphere spline model

We also timed the algorithm when computing the gravitational model using a local cubed sphere spline model [3]. The timing results are given in Table 2.

Degree	CPU Time (s)	Number of function calls
41	3.2E-02	6160
70	3.2E-02	6160

Table 2
CPU time and number of function calls to full gravitational model when computing a orbit of 86000 seconds using a local cubed sphere model.

6.2.3 Current (indirect, crude) estimate of timing

We tested the speed of our code on a sample NOAA satellite orbit, for which we received timing results using the SPEPH code. Since the SPEPH code and

our code ran on different machines, we were able only to estimate the timing using the following approach. We were given the speed of SPEPH code for gravitational models with $N = 36$ and $N = 70$ order and degree. Assuming that timing captures the cost of integration and access to the gravity model and assuming that the cost of integration is the same for both models, we write

$$t_{total}^{(deg)} = t_{code} + t_{model}^{(deg)},$$

where t_{code} is the time of integration and $t_{model}^{(deg)}$ is the time of access to the model. We further assume that

$$t_{model}^{(70)}/t_{model}^{(36)} = (70/36)^2 \approx 3.78,$$

since the cost of access to the spherical harmonic model grows quadratically with its order and degree. Given this information for SPEPH code and timing our code we obtain the following comparison by solving the above equations

$$\begin{aligned}
t_{model}^{(36)} &= \frac{t_{total}^{(70)} - t_{total}^{(36)}}{3.78 - 1} \approx 0.36 \cdot (t_{total}^{(70)} - t_{total}^{(36)}) \\
t_{model}^{(70)} &= 3.78 \cdot t_{model}^{(36)} \approx 1.36 \cdot (t_{total}^{(70)} - t_{total}^{(36)}) \\
t_{code} &= t_{total}^{(36)} - t_{model}^{(36)}.
\end{aligned}$$

Using the values from timing SPEPH, $t_{total}^{(36)} = 5.21$ and $t_{total}^{(70)} = 16.36$, we get

$$\begin{aligned}
t_{model}^{(36)} &\approx 4.0 \\
t_{model}^{(70)} &\approx 15.12 \\
t_{code} &\approx 1.21 \\
\frac{t_{model}^{(70)}}{t_{code}} &\approx 12.5
\end{aligned}$$

For our code we have $t_{total}^{(36)} = 6.3$ and $t_{total}^{(70)} = 18.6$, and we get

$$\begin{aligned}
t_{model}^{(36)} &\approx 4.4 \\
t_{model}^{(70)} &\approx 16.63 \\
t_{code} &\approx 1.9 \\
\frac{t_{model}^{(70)}}{t_{code}} &\approx 8.75
\end{aligned}$$

Thus, currently, our implicit code appears to be about 30% slower than the explicit SPEPH code (if the indirect estimate above is correct). Also, it appears that the computation of the spherical harmonics has been accelerated in the current SPEPH code in comparison with its predecessor since the tests of $6-7$

years ago were done. For this reason, the local gravity model developed in [3] is faster than the spherical harmonic model by a lower factor, perhaps only a factor of 10 rather than 40 as before. However, this may be changed to higher performance factors so that the local model becomes faster since it is a simple matter of trading speed vs memory (memory is no longer a problem for the sizes needed for these models). We also need to accelerate our code to match and, hopefully, exceed the speed of the current SPEPH code. Note that the our code is A-stable and symplectic (if need be). The code is currently being tested for accuracy by Brandon Jones and appears to match the results of the codes used in CU Aerospace Department.

7 Conclusions

We have constructed an implicit, symplectic integrator that has speed comparable to the explicit multistep integrator currently used for orbit computation. In combination with a local model for gravity it achieves a factor of 8.3 improvement in speed compared to using a global gravity model.

References

[1] Bradley K. Alpert. Hybrid Gauss-trapezoidal quadrature rules. *SIAM J. Sci. Comput.*, 20(5):1551–1584, 1999.

[2] G. Beylkin, N. Coult, and M. J. Mohlenkamp. Fast spectral projection algorithms for density-matrix computations. *Journal of Computational Physics*, 152(1):32–54, 1999.

[3] G. Beylkin and R. Cramer. Toward multiresolution estimation and efficient representation of gravitational fields. *Celestial Mechanics and Dynamical Astronomy*, 84(1):87–104, 2002.

[4] G. Beylkin and L. Monzón. On generalized Gaussian quadratures for exponentials and their applications. *Appl. Comput. Harmon. Anal.*, 12(3):332–373, 2002.

[5] G. Beylkin and K. Sandberg. Wave propagation using bases for bandlimited functions. *Wave Motion*, 41(3):263–291, 2005.

[6] K. Dekker and J.G. Verwer. *Stability of the Runge-Kutta methods for stiff nonlinear differential equations*. North-Holland, Amsterdam, 1984.

[7] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT*, 40(2):241–266, 2000.

[8] D. Gottlieb and S. A. Orszag. *Numerical analysis of spectral methods: theory and applications*. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1977. CBMS-NSF Regional Conference Series in Applied Mathematics, No. 26.

[9] J. Huang, J. Jia, and M. Minion. Accelerating the convergence of spectral deferred correction methods. *J. Comput. Phys.*, 214(2):633–656, 2006.

[10] A. Iserles. *A first course in the numerical analysis of differential equations*. Cambridge University Press, 1996.

[11] H. J. Landau and H. O. Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty II. *Bell System Tech. J.*, 40:65–84, 1961.

[12] H. J. Landau and H. O. Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty III. *Bell System Tech. J.*, 41:1295–1336, 1962.

[13] J.M. Sanz-Serna. Runge-Kutta schemes for Hamiltonian systems. *BIT*, v. 28:877–883, 1988.

[14] D. Slepian. Prolate spheroidal wave functions, Fourier analysis and uncertainty IV. Extensions to many dimensions; generalized prolate spheroidal functions. *Bell System Tech. J.*, 43:3009–3057, 1964.

[15] D. Slepian. Prolate spheroidal wave functions, Fourier analysis and uncertainty V. The discrete case. *Bell System Tech. J.*, 57:1371–1430, 1978.

[16] D. Slepian. Some comments on Fourier analysis, uncertainty and modeling. *SIAM Review*, 25(3):379–393, 1983.

[17] D. Slepian and H. O. Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty I. *Bell System Tech. J.*, 40:43–63, 1961.

[18] H. Xiao, V. Rokhlin, and N. Yarvin. Prolate spheroidal wavefunctions, quadrature and interpolation. *Inverse Problems*, 17(4):805–838, 2001.